AMENDMENTS TO THE CLAIMS

1-19 (Cancelled)

20. (New) A method comprising:

detecting a stall in an execution stage of a processor;

generating a neutral instruction;

providing said neutral instruction to said execution; and

executing said neutral instruction to ascertain an architectural state value for said processor.

21. (New) The method of claim 20 wherein said neutral instruction is generated when a plurality of instructions are generated by a compiler.

22. (New) The method of claim 20 wherein said neutral instruction is generated by a No-operation (NOP) pseudo-random generator.

23. (New) The method of claim 22 wherein the execution of said neutral instruction causes said processor to access a value stored in a register in said processor.

24. (New) The method of claim 20 wherein the execution of said neutral instruction causes said processor to access a value stored in a register in said processor.

25. (New) The method of claim 20 wherein said neutral instruction is generated by a post-processor device.

26. (New) A system comprising:

stall logic coupled to an execution stage of a processor to detect a stall in said execution; and

comparison logic coupled to said execution stage, wherein upon occurrence of the stall said execution stage is to execute a neutral instruction to ascertain an architectural state value for said processor.

27. (New) The system of claim 26 wherein said neutral instruction is generated by a compiler.

28. (New) The system of claim 26 further comprising:

a No-operation (NOP) pseudo-random generator coupled to the execution unit of said processor to generate said neutral instruction.

29. (New) The system of claim 28 wherein the processor includes a register and the execution of said neutral instruction causes said processor to access a value stored in the register in said processor.

30. (New) The system of claim 29 wherein said neutral instruction includes ORing the contents of said register with itself.

31. (New) The system of claim 29 wherein said neutral instruction includes ANDing the contents of said register with all binary 1 values.

32. (New) The system of claim 29 wherein said neutral instruction includes ORing the contents of said register with all binary 0 values.

33. (New) A set of instructions residing in a storage medium, said set of instructions capable of being executed in an execution stage by a processor for implementing a method to test the processor, the method comprising:

detecting a stall in an execution stage of a processor;

generating a neutral instruction;

providing said neutral instruction to said execution; and

executing said neutral instruction to ascertain an architectural state value for said processor.

34. (New) The set of instructions of claim 33 wherein in said method said neutral instruction is generated when a plurality of instructions are generated by a compiler.

35. (New) The set of instructions of claim 33 wherein in said method said neutral instruction is generated by a No-operation (NOP) pseudo-random generator.

36. (New) The set of instructions of claim 35 wherein in said method the execution of said neutral instruction causes said processor to access a value stored in a register in said processor.

37. (New) The set of instructions of claim 33 wherein in said method the execution of said neutral instruction causes said processor to access a value stored in a register in said processor.

38. (New) The set of instructions of claim 33 wherein in said method said neutral instruction is generated by a post-processor device.